

Quantitative Abstraction Theory

Chris Thornton

Cognitive and Computing Sciences, University of Sussex
Falmer, Brighton, BN1 9QH, UK
chris.thornton@firenet.uk.com

Abstract

A quantitative theory of abstraction is presented. The central feature of this is a growth formula defining the number of abstractions which may be formed by an individual agent in a given context. Implications of the theory for artificial intelligence and cognitive psychology are explored. Its possible applications to the issue of implicit v. explicit learning are also discussed.

1 Introduction

Abstraction has long been assumed to be a key process in cognition. And though it has never been given a generic specification, philosophers since the time of Aristotle have been willing to accord it a central role. In Aristotle's case (1), and later in Locke's (2), the process was seen as lying at the heart of the problem of 'universals'.¹ More recently, artificial intelligence researchers have carried on the tradition, letting abstraction take the strain in models of search, problem solving, theorem proving, planning, reasoning and programming [c.f. 3, Gunchiglia and Walsh, 1990, 4, 5].

That such an important process has never been put on a formal, theoretical footing is a little odd. It may be that it is regarded as too obvious and straightforward to require formuleaic treatment. And, certainly, there has been little dispute down the ages about the nature of the process itself. Accounts of the process have tended to show strong commonalities. For example, consider Hume's description from 'The Essay' (6).

'Tis evident that in forming most of our general ideas, if not all of them, we abstract from every particular degree of quantity and quality, and that an object ceases not to be of any particular species on account of every small alteration in its extension, duration and other properties. (pp.16-7).

Hume saw abstraction, then, in terms of the filtering-away of information of specifics, with the aim of extracting content or meaning. Aristotle also saw it this way. So too did Locke, the philosopher perhaps most strongly associated with the idea that universals are derived by abstraction from empirical data. Locke, in fact, deemed abstraction to be the leaving out of particular circumstances of time and place.

But even a brief examination of these portrayals of the process reveal underlying inconsistencies, contradictions and ambiguities. Philosophers may agree that abstraction involves the elimination of relatively specific information. But they are less clear how

¹One characterisation of the difference between Plato's and Aristotle's views on the derivation of universal truths was that Plato saw them as coming 'from above' while Aristotle saw them as coming 'from below'.

the specificity is to be measured or how the information is to be represented. In fact, any attempt to specify what is to be eliminated tends to fall foul of counter-examples. Locke's notion, for instance, that all factors relating to time and place should be eliminated would seem quite inappropriate in the case of abstraction applied to higher-level concepts, such as might relate to physical beauty for example.

Suffice it to say, then, that philosophical accounts of the process of abstraction are characteristically *pre-theoretical*. They assume the existence of a well-defined, shared meaning for the term. Since this does not exist, these accounts lack precision and are insufficient for the mechanistic and programmatic purposes of AI.

No surprise, then, that AI projects which attempt to harness the power of abstraction in a particular problem domain typically start by providing a mechanistic definition of the process (cf. Gunchiglia and Walsh, 1990). Clearly, AI researchers are aware of the fact that abstraction has no *generic* specification and that they cannot hope to make use of it without first providing a working definition.

The formulation of a generic specification for the process is a worthy goal, then, which might yield benefits right across the landscape connecting cognitive science to epistemology. It could provide a generic basis for the diversity of abstraction-using AI techniques. It might also provide a means of integrating abstraction-related ideas arising in different areas. Possibly, it might also help to fertilise new techniques for exploiting abstraction within cognition. Last but not least, it would help to further the theoretical development of artificial intelligence.

But while the present paper takes this ambitious goal as its general context, it makes no claim to reach the target or even to approach it very closely. Rather, it addresses the special problem of *abstraction quantification*. The paper shows, in particular, how we may calculate the number of abstractions which may be generated by an individual agent in a particular context. In so doing, it develops and uses a partial formalisation of the process itself. This turns out to have a number of practical and explanatory applications within AI and in related areas such as cognitive psychology. There is also the hint of a new angle on the longstanding problem of universals.

2 Derivation of the theory

Informal characterisations of abstraction (such as Hume's) normally focus on the reductive aspects of the process, i.e., the way in which relatively specific information is eliminated. But the process may also be characterised in terms of its *constructive* function. An abstraction is necessarily an abstraction *of* something. In essence, then, it is an identification of a phenomenon — an object, process or property of the abstracting agent's world. At the point of construction, the constituents must be already available. We may view abstraction therefore not in terms of the elimination of irrelevant components, but in terms of the selection and combination of relevant constituents.

The advantage of the constructive interpretation is that it opens up the possibility for quantitative analysis. Since the result of any act of abstraction is the identification of a new phenomenon embodying some combination of currently identified phenomena, we can use combinatorial reasoning to determine the number of abstractions a given agent can form starting from a base of primitive identifications.

However, there are several complications to take into account. The number of possible abstractions might seem simply to be the number of ways in which the elements of the base set may be combined. But this is not quite correct. Each new abstraction identifies a new phenomenon and thus becomes a potential constituent in a further abstraction. The

process, then, is inherently recursive. The analysis should take account of this.

Also of importance is the fact that there are two quite different ways in which identifications may be combined to form a new abstraction. First, there is the process of *composition* in which identified phenomena are combined together as parts to form a new whole. Second, there is the process of *classification* in which identified phenomena are gathered together (as whole elements) into a class of alternatives. (In AI terms, the former is construction using PARTOF relationships and the latter is constructing using ISA relationships.) Every possible subgroup of identifications is a candidate for both processes. Thus, starting from any base set, we may derive a set of abstractions by treating each possible subgroup as (a) a composite and (b) a class.

The general idea is visualised in Figure 1. Here the base set of identifications is labelled P_0 . From P_0 , we obtain P_1 : each identification in this set is an abstraction derived by applying composition or classification to a subset of P_0 . Treating P_1 as the base set permits the derivation of a set P_2 in which each phenomenon is the result of classification or composition applied to a subset of P_1 . Treating P_2 as the base set permits the derivation of the set P_3 and so on. In this manner, we can go on to derive P_4 , P_5 , P_6 etc.

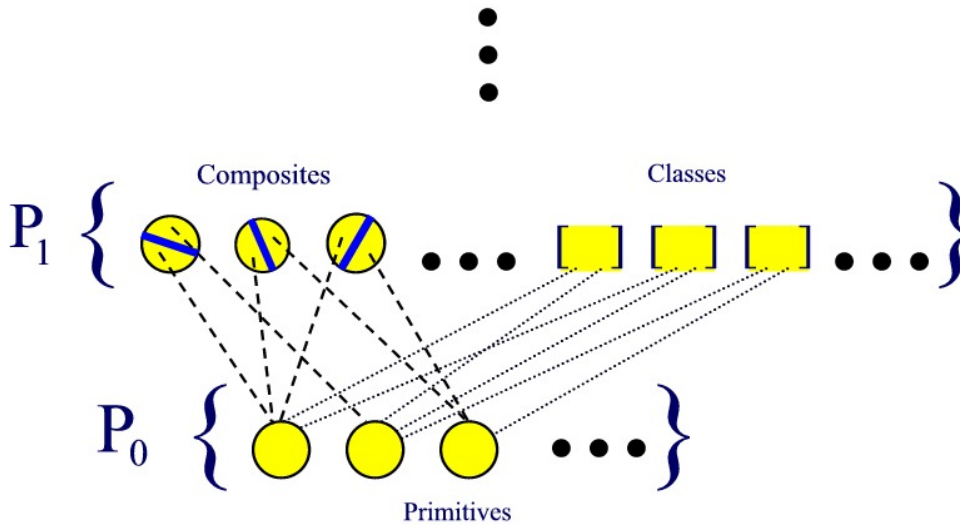


Figure 1: Abstraction tree.

The diagram portrays the generation of possible abstractions, starting from a set of primitives. The dynamics of this are characteristically constructive. But the eliminative aspect should also be apparent. In the case of both composition and classification, a set of elements is reidentified as a single entity. Information relating to the elements themselves is effectively eliminated. But the result is achieved in two different ways. In the case of composition, the elements become the component parts of a new whole. In classification,

the elements become alternative manifestations of a single identity.

3 Complexity

Applied recursively to a base set of identifications, the two forms of abstraction lead to an infinite hierarchy of constructs. The number of nodes in this hierarchy expands rapidly as we move upwards from level to level. Let us say there are n nodes in a particular layer. Then we would expect the number of nodes in the layer above to be

$$2(2^n)$$

since the number of possible combinations of n objects is 2^n , and the process generates two nodes for each combination. However, we must also account for the fact that some of these nodes are redundant. Clustering applied to all possible classes of a set of objects is redundant, since any class obtained must be identical in object membership to one of the original classes. By the same token, abstractions involving classes composed of classes are redundant. Thus we need to discount the nodes which result from classification applied *purely* to classes.

Exactly half of the n nodes will be classes. Therefore we should subtract $2^{\frac{n}{2}}$. The revised formula for the number of nodes then becomes

$$2(2^n) - 2^{\frac{n}{2}}$$

It might seem that a further modification should be made to take account of the fact that exactly n of the 2^n possible combinations are singleton sets, i.e., they simply yield ‘copies’ of nodes at the layer below. (We might discount these nodes by subtracting $2n$.) However, since it is possible in principle for abstractions to be constructed out of nodes at different levels in the tree, it simplifies matters if we allow the singleton sets to remain. This way, every level of the tree contains a copy of every node at every level below and the possibility of cross-level abstractions is automatically taken into account.

To render the formula in a recursive form is now straightforward. If n_0 is set equal to the number of basic elements, the number of nodes n_i represented at the i ’th level of the hierarchy may then be calculated using the following recursive formula.

$$n_{i+1} = 2(2^{n_i}) - 2^{\frac{n_i}{2}}$$

4 Significant abstractions

The growth formula reveals the exponential cost of abstraction formulation. But it applies specifically to the case of exhaustive (i.e., unrestrained) abstraction rather than to abstraction in practice. The difference is significant. Agents which form abstractions in a realistic situation are surely unlikely to do so *exhaustively*. More likely, they will aim to ensure that abstractions match up to reality. This will involve making sure that any identifications formed are literally *significant*, i.e., identify real and salient phenomena. (Concerns about the inaccessibility and/or implausibility of objective reality are ignored here.) The net effect is that the set of *significant abstractions* for any particular individual is likely to be a small subset of the total set of possible abstractions.

But although the costs of in-practice abstraction may be lower than the growth formula suggests, they will not be as low as we might hope. They will, after all, include the

Thornton

costs of carrying out the ‘reality check’, i.e., whatever operation is required to ensure that abstractions match reality. In the case of classification, this may involve nothing more than making observations about similarities among the relevant class members. But in the case of compositional abstraction, the resulting construct is only valid if the elements fit together in the right way, i.e., only if they have the right relationships. Thus the formation of compositional abstractions always involves the identification, by the abstracting agent, of the relevant relationship. There is evidence to suggest that in the worst case this may be an infinitely complex task (7).

5 Types and tokens

Any abstraction whose structure (in the hierarchy) is not, at any stage, mediated by classification (i.e., whose roots do not go back through any class nodes) has only one, possible grounding in basic elements. In the perception of the agent, there is only one way that it exists. As a conceptualisation, then, the abstraction constitutes a *token*. In contrast, any abstraction whose derivation is mediated by classification (i.e., whose roots do go back through class nodes) identifies a phenomenon with more than one possible grounding in basic elements. With respect to the given set, the latter constitutes a *type*, since it effectively stands for more than one combination of elements.

The theory thus gives a formal meaning to the long-standing distinction between types and tokens. But note how it upgrades the idea from a simple dichotomy into a continuous dimension. As noted, the roots of a type node must go back through one or more class nodes. But there can be more or less of these. And they may appear higher or lower in the tree. Thus, the ‘typeness’ of a phenomenon is not a black-and-white issue. Rather, it is a matter of (2-dimensional) degree.

How then should we properly render the distinction between types and tokens? A simple approach might be to treat *every* phenomenon as a type, and to say that the ‘typeness’ of a particular phenomenon is just the size of its extension — the number of ways in which it can exist. A token could then be thought of as a type with an extension of one.

An alternative would be to treat an identification as a type only if its class nodes are sufficiently close to the surface, i.e., appear sufficiently high in the relevant abstraction construct. There might be problems in identifying a suitable cutoff point. But the approach has its attractions. It would, for example, avoid the necessity of treating the individual called Fred Bloggs as a ‘type’ simply on the grounds that he may consist, at any one time, of quite different arrangements of quantum states. The roots of ‘Fred Bloggs’ may go back through class nodes, we could argue, but they are at too great a depth to be treated as significant.

Perhaps the best approach is simply to accept that the traditional type/token terminology over-simplifies reality. The logical structure of abstraction means that the size and character of a phenomenon’s extension may vary in a range of ways. Therefore there can be no hard and fast distinction between types and tokens.

6 Other applications of the theory

The growth formula and its underlying principles provides a definition of the term ‘abstraction’ and a means of estimating the number of abstractions which may be formed by an individual agent in a given situation. It provides a *quantitative* theory of abstraction rather than a qualitative one, since it says nothing about what abstractions will actually

consist of, except that they will involve the combination of certain elements. It also says nothing about the way in which the process works or why it is required.

The theory can be applied to natural agents and even, in principle, to human subjects. But here there is always the problem of identifying the set of basic identifications upon which abstraction may build. Without a specification for this set, the growth formula cannot be applied and the rest of the theory becomes inoperable. In some cases, it may be feasible to treat an agent's sensory stimuli as the set of fundamental identifications of phenomena. (Certainly, there could be no more basic set of primitives than this.) But in practice the approach still raises horrific problems of enumeration.

More practical are applications which focus on artificial agents, particularly when these are hand-designed. Very often, the basic set of environmental objects with which a designed agent engages can be simply read-off the design. The derivation of the potential abstraction tree is then straightforward.

In some cases, it may be useful to map out an agent's total abstraction set simply as a means of evaluating its possible, representational trajectories (cf. 8). This might also provide the basis for an evaluation of the agent's conceptual *adventurousness*. Its relative penetration of the total abstraction set — the ratio between the number of developed and potential abstractions — summarises the degree to which the agent has fleshed-out the potential conceptualisations of its environment. Relative penetration might then become a kind of conceptual 'horse-power' rating for artificial agents. (This is not completely satisfactory, however, since the total abstraction set will normally be significantly larger than the set of significant abstractions.)

7 Representation and behaviour

To some degree, the theory may also be used to make judgements about the representational behaviour of agents. An agent's total abstraction set includes all the phenomena that the agent is capable of identifying (including ones that do not actually exist). Putting this in representational terms, we would say that the abstraction tree identifies the complete set of phenomena that the agent is capable of representing, as well as the logical dependencies between them.

Thus, if a particular phenomenon does not appear within an agent's total abstraction set, we know that the agent *cannot* form a representation for that phenomenon. It may be unable to form a particular representation for other reasons. But the absence of the phenomenon from the abstraction set shows that it cannot do so *in principle*. This might become an issue of importance, for instance, if an attempt were being made to build an agent that would acquire the ability to behave contingently with respect to a property of the world that it was unable to represent.

Imagine for example that the aim is to construct a simple, mobile agent which will acquire the ability to approach smooth objects but not spikey ones. Regardless of any efforts made, the experiment will necessarily fail if the phenomenon 'smooth object' has no representation within the agent's abstraction tree.

But the representational implications of the theory only go so far. It allows one to calculate what is contained within a particular abstraction set and thus what a particular agent is and is not capable of representing. However, it says nothing about what a particular representation will consist of or how it will be constructed. (This is really just the same point as was made above: the theory does not specify what an abstraction will consist of, merely that it must combine certain elements.)

Thornton

Furthermore, no claim is made that agents will representationally reproduce the *structure* of abstraction trees. Indeed, it is apparent that areas of research interested in learning and behaviour acquisition (whether motivated by a representational interests or not) show little sign of devising methods which generate abstraction trees, or anything like them. If anything, the reverse is the case (cf. 9). The evidence is that insofar as contemporary artificial agents may be said to build representations *at all*, these do not resemble abstraction trees.

On the other hand, it is noticeable that the very same areas of research tend to divide attention between classification (class-forming) methods and compositional methods. In other words, they may be viewed as dividing attention between the two fundamental processes of abstraction. This is perhaps most noticeable in machine learning, which is broadly divided up into a subfield focussing on statistical classification methods (similarity-based learning) and a subfield focussing on compositional or relational methods (discovery, analogy, inductive logic programming etc.) (10)

8 Explicit and implicit learning

Perhaps the most fruitful area for applications of the theory is that of cognitive psychology. A lively debate in this area involves the problem of explicit v. implicit learning. In part, this is concerned with the question of whether knowledge is stored in an abstract or specific (i.e., instance-based) form. It also focusses on the degree to which knowledge is the result of conscious or unconscious processes. (In some sense, the two parts of the debate are really one, with the former focussing on static issues and the latter on dynamic.)

Traditionally, the implicit/explicit issue has been researched using experiments in which human subjects are either taught, or exposed to strings generated by an artificial grammar. By evaluating the subject's ability to classify test cases, or to transfer knowledge from one grammar to another, deductions are attempted showing the degree to which abstractions have or have not resulted from implicit learning processes.

The seminal work in this area was performed by Reber (11) and it was his main conclusion that unconscious (i.e., implicit) processes of learning could produce internal abstractions with the same functional properties as those acquired through explicit tuition. The implications that Reber drew from his results have been widely questioned, with objections often focussing on the fuzziness of the supposed dichotomy between abstract and specific knowledge.

All business as usual, perhaps. But from the point of view of quantitative abstraction theory, it begins to look as if the problem here, as with the dilapidated type/token distinction, may really be the result of the attempt to apply an over-simplified, black-and-white conceptualisation to what is in reality a complicated, multi-dimensional issue.

According to the theory, abstractions have a logical structure which may be *arbitrarily* deep. The derivation of new abstractions, whether classificatory or compositional, may proceed at any level within the tree of existing abstractions. On this view, it makes no sense to classify new knowledge as either abstract or specific. Rather, it should be identified as having a particular *level* of abstraction.

Similar remarks can be made with respect to conscious v. unconscious processing. Assuming that the level of 'consciousness' inherent in cognitive processing is a function of the abstractness of the entities over which it is applied, we can apply the continuity upgrade to the conscious/unconscious 'dichotomy' too. On the basis of the assumption stated, we can treat the issue of the consciousness of processing as a matter of degree, and

judge any specific cases according to elevation in the relevant abstraction tree.

9 Concluding comment

It would be an interesting project to determine how many of the unresolved issues surrounding the question of implicit v. explicit learning would evaporate in the presence of suitable enhancements in the terminology. The project would certainly be approved by Alan Newell who nearly forty years ago admonished psychologists for their use of simplistic, binary opposites in their conceptualisation of cognitive function (12). Rather obviously, Newell's criticisms have had a limited impact. In fact, the implicit v. explicit debate gained its principal momentum a full decade *after* Newell's publication. But the fact that cognitive psychologists still adhere to black-and-white concepts may be due to the fact that workable replacements have yet to be provided. In this context, the limited but concrete contribution of the present theory may have a worthwhile future.

References

- [1] Aristotle, (1984). In J. Barnes (Ed.), *The Complete Works of Aristotle*. Princeton: Princeton University Press.
- [2] Locke, J. (1690/1978). In A. Pringle-Pattison (Ed.), *An Essay concerning Human Understanding* (first pub. 1690). Sussex: Harvester.
- [3] Plaisted, D. (1981). Theorem proving with abstraction. *Artificial Intelligence*, 16 (pp. 47-108). 1.
- [4] Sacerdoti, E. (1974). Planning in a hierarchy of abstraction spaces. *Artificial Intelligence*, 5 (pp. 115-135). 2.
- [5] Knoblock, C. (1994). Automatically generating abstractions for planning. *Artificial Intelligence*, 68, No. 2.
- [6] Hume, D. (1740). *A Treatise of Human Nature* (second edition). Oxford University Press.
- [7] Clark, A. and Thornton, C. (1997). Trading spaces: computation, representation and the limits of uninformed learning. *Behaviour and Brain Sciences*, 20 (pp. 57-90). Cambridge University Press.
- [8] Clark, A. and Karmiloff-Smith, A. (1993). The cognizer's innards: a psychological and philosophical perspective on the development of thought. *Mind and Language*, 8.
- [9] Brooks, R. (1991). Intelligence without reason. *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence* (pp. 569-595). San Mateo, California: Morgan Kaufman.
- [10] Russell, S. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach* (Second Edition). Prentice Hall.
- [11] Reber, A. (1967). Implicit learning of artificial grammars. *Journal of Verbal Learning and Verbal Behaviour*, 5 (pp. 855-863).

Thornton

- [12] Newell, A. (1965). Limitations of the current stock of ideas for problem solving.
In A. Kent and O. Taulbee (Eds.), *Conference on Electronic Information Handling*.
Washington, D.C.: Spartan.